Cenni di Teoria dell'Informazione

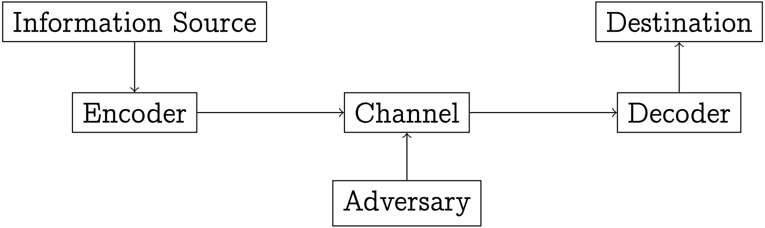
Crittografia

Luciano Margara

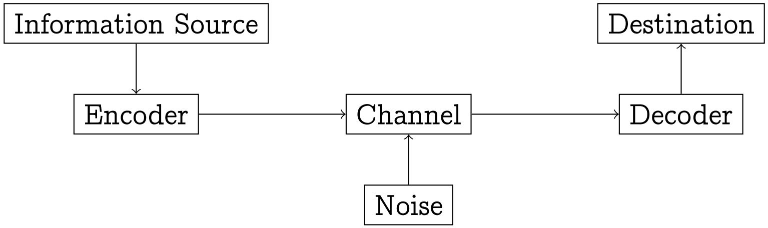
Unibo

2022

Crittografia: Scenario



Teoria dell'Informazione: Scenario



Teoria dell'Informazione Vs Crittografia

Noise $\longrightarrow$ Teoria dell'Informazione
Adversary $\longrightarrow$ Crittografia

Encoder

Encoder = "Compressione" + "Protezione"

Compressione

Eliminare tutto ciò che non è "informazione"

Compressione stringhe random

```
GeneraStringaBinaria[lunghezza_, prob_] :=
 Module[{r, i, risultato},
  risultato = "";
  For[i = 1, i <= lunghezza, i ++,
   r = RandomReal[];
   If[r ≤ prob,
    risultato = StringJoin[risultato, "0"],
    risultato = StringJoin[risultato, "1"]
   ];
  ];
  Return[risultato];
 ]
```

```
GeneraStringaBinaria[20, 0.2]
```

```
11011111011111111100
```

Compressione stringhe random

```
s = GeneraStringaBinaria[10, 0.5]
```

1010111001

```
c = Compress[s]
```

1:eJxTTMoPCuZiYGAwNABCQ0MDA0MAJcQDiA==

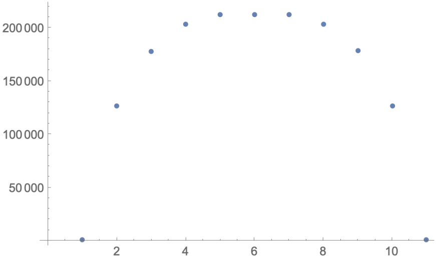
```
s = Uncompress[c]
```

1010111001

Compressione stringhe random

```
t = Table[StringLength[Compress[GeneraStringaBinaria[1 000 000, i]]], {i, 0, 1, .1}]
{1346, 126 654, 178 186, 203 874, 212 598, 212 410, 212 586, 203 738, 178 618, 126 406, 1346}
```

```
ListPlot[t]
```



Protezione

Aggiungere "rivestimento" che protegga dall'errore (noise)

Domande

Che cosa è l'Informazione?
Come si misura?
Come si comprime?
Quanto si comprime?
Come si protegge?

Sorgente di informazione

- Alfabeto: $\Sigma = \{a_1, a_2, \dots, a_M\}$
- Stringhe finite:
  $\Sigma^* = \{\text{sequenze finite di caratteri in } \Sigma\}$
- Distribuzione di Probabilità: $P = \{p_1, p_2, \dots, p_M\}$

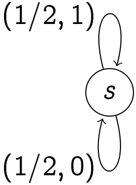
Esempio

- $\Sigma = \{0, 1\}$
- $\Sigma^* = \{\text{sequenze binarie finite}\}$
- Distribuzione di Probabilità: $P = \{1/2, 1/2\}$

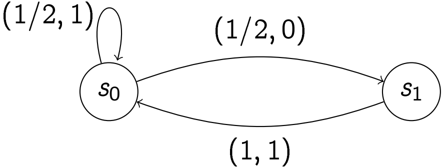
Sorgente di Informazione

Modello semplice: sequenza di variabili aleatorie discrete identicamente distribuite e a due a due indipendenti.
Modelli più complicati: Catene di Markov, etc etc.

Catene di Markov



Catene di Markov



Intuitivamente

Informazione
Incertezza
Entropia

$H(p_1, \ldots, p_M)$

Sia $H(M) = H(1/M, \ldots, 1/M)$

- $M < L \implies H(M) < H(L)$
- Sorgenti indipendenti $\implies H(M\,L) = H(M) + H(L)$

$H(p_1, \ldots, p_M)$

- Grouping axiom:

$$H(p_1, \ldots, p_r, p_{r+1}, \ldots, p_M) =$$
$$H\left(\sum_{i=1}^{r} p_i, \sum_{i=r+1}^{M} p_i\right) +$$
$$\sum_{i=1}^{r} p_i \, H\left(\frac{p_1}{\sum_{i=1}^{r} p_i}, \ldots, \frac{p_r}{\sum_{i=1}^{r} p_i}\right) +$$
$$\sum_{i=r+1}^{M} p_i \, H\left(\frac{p_{r+1}}{\sum_{i=r+1}^{M} p_i}, \ldots, \frac{p_M}{\sum_{i=r+1}^{M} p_i}\right) +$$

$H(1/6, 1/6, 1/6, 1/6, 1/6, 1/6)$

$$H(\frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}, \frac{1}{6}) =$$
$$H(\frac{1}{3}, \frac{2}{3}) +$$
$$\frac{1}{3} H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{2}{3} H\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, \frac{1}{4}\right)$$

$H(p_1, \ldots, p_M)$

- H continua

Entropia

$$H(S) = -C \sum_{i=1}^{M} p_i \log(p_i)$$

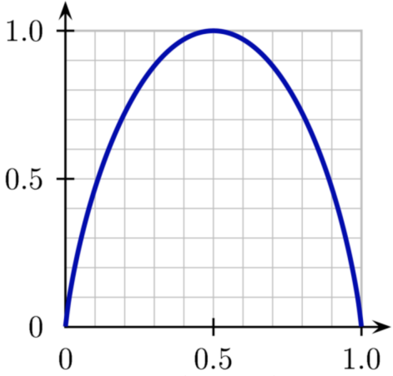
Entropia

$$H(S) = -\sum_{i=1}^{M} p_i \log_2(p_i)$$

$$H_D(S) = -\sum_{i=1}^{M} p_i \log_D(p_i)$$

Entropia del lancio di una moneta

$$H(p, 1-p) = -p\log(p) - (1-p)\log(1-p)$$



Bit

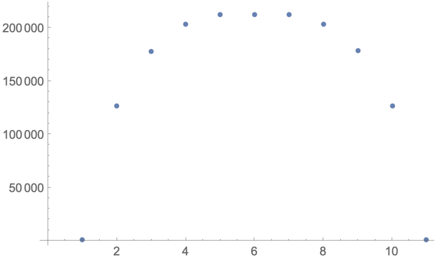
Unità di misura di H è il bit

Compressione Vs Entropia

```
t = Table[StringLength[Compress[GeneraStringaBinaria[1 000 000, i]]], {i, 0, 1, .1}]
```

{1346, 126 654, 178 186, 203 874, 212 598, 212 410, 212 586, 203 738, 178 618, 126 406, 1346}

```
ListPlot[t]
```



Codici non unicamente decifrabili

$$\begin{array}{ccc} a_1 & \rightarrow & 0 \\ a_2 & \rightarrow & 010 \\ a_3 & \rightarrow & 01 \\ a_4 & \rightarrow & 10 \end{array}$$

$$\begin{array}{ccc} a_2 & \rightarrow & 010 \\ a_3\, a_1 & \rightarrow & 010 \\ a_1\, a_4 & \rightarrow & 010 \end{array}$$

Codici unicamente decifrabili

Un codice è unicamente decifrabile se ogni sequenza di caratteri di codice corrisponde ad al massimo un messaggio

Codici istantanei

Un codice è istantaneo se nessuna parola di codice è prefisso di un'altra parola di codice

Un codice istantaneo è unicamente decifrabile

Codici

- Alfabeto: $\Sigma = \{a_1, a_2, \ldots, a_M\}$
- Distribuzione di Probabilità: $P = \{p_1, p_2, \ldots, p_M\}$
- Parole di codice in base D unicamente decifrabile: $W = \{w_1, w_2, \ldots, w_M\}$
- Lunghezze delle parole di codice: $n_i = |w_i|, \ 1 \le i \le M$

Codici istantanei

Un codice istantaneo in base D con parole di codice $w_1, \ldots, w_M$ di lunghezza n_i esiste se e solo se

$$\sum_{i=1}^{M} D^{-n_i} \leq 1$$

Teorema

Sia $\mu = \sum_{i=1}^{M} p_i \, n_i$ la lunghezza media delle parole di codice. Allora

$$\mu \geq H_D(p_1, p_2, \ldots, p_M)$$

e

$$\mu = H_D(p_1, p_2, \ldots, p_M) \iff p_i = D^{-n_i}$$

Teorema

Esiste un codi istantaneo $W = \{w_1, w_2, \ldots, w_M\}$ con lunghezze delle parole $n_i = |w_i|$ tale per cui

$$H_D(p_1, p_2, \ldots, p_M) \leq \mu < H_D(p_1, p_2, \ldots, p_M) + 1$$

dove $\mu = \sum_{i=1}^{M} p_i \, n_i$

Codice assolutamente ottimo

Σ	p_i	w_i
a_1	$1/2$	0
a_2	$1/4$	10
a_3	$1/8$	110
a_4	$1/8$	111

$$H(1/2, 1/4, 1/8, 1/8) = \mu = \frac{7}{4}$$

Estensione della sorgente

Σ	p_i	w_i
a_1	$3/4$	0
a_2	$1/4$	1

$$H(3/4, 1/4) = 0.811278 \quad \mu = 1$$

Σ^2	$p_i p_j$	w_{ij}
$a_1 a_1$	$9/16$	0
$a_1 a_2$	$3/16$	10
$a_2 a_1$	$3/16$	110
$a_2 a_2$	$1/16$	111

$$H(9/16, 3/16, 3/16, 1/16) = 1.62 \quad \mu = 27/16 = 1.68$$

$$H(9/16, 3/16, 3/16, 1/16)/2 = 0.81 \quad \mu/2 = 27/32 = 0.84$$

In pratica

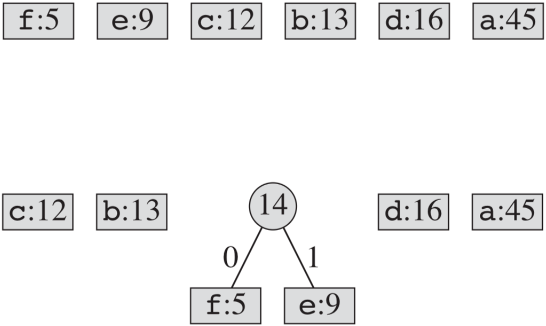
Non è nota la distribuzione di probabilità associata alla sorgente di informazione, ma è noto il file da comprimere. Da quello è possibile risalire alla distribuzione statistica dei caratteri, delle coppie di caratteri, delle terne, etc etc

Esempio

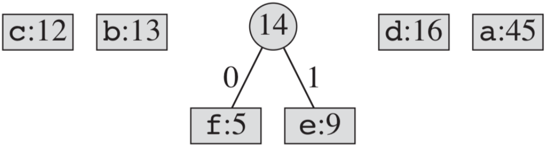
- $\Sigma = \{a, b, c, d, e, f\}$
- $|s| = 100.000$
- Frequenze (in migliaia): $a : 45$, $b : 13$, $c : 12$, $d : 16$, $e : 9$, $f : 5$

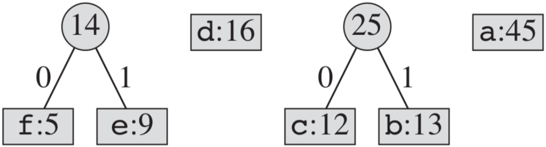
Codici di Huffman

Esempio

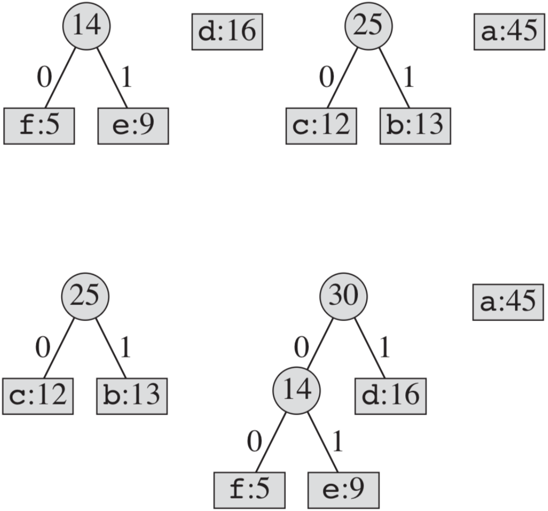


Esempio

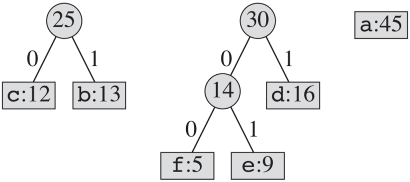


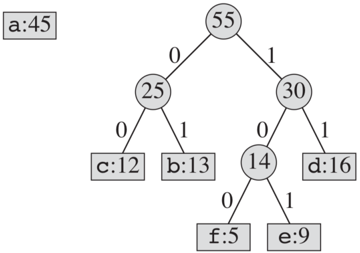


Esempio

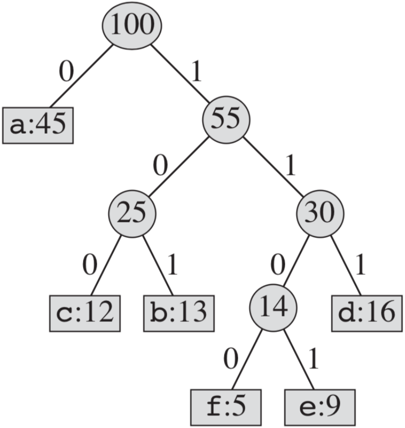

14
0
1
f:5
e:9
d:16
25
0
1
c:12
b:13
a:45
25
0
1
c:12
b:13
30
0
1
14
0
1
f:5
e:9
d:16
a:45

Esempio


25
0
1
c:12
b:13
30
0
1
14
d:16
0
1
f:5
e:9
a:45


a:45
55
0
1
25
30
0
1
0
1
c:12
b:13
14
d:16
0
1
f:5
e:9

Esempio



Codifica

simbolo	Huffman	standard
a	0	000
b	101	001
c	100	010
d	111	011
e	1101	100
f	1100	101

Codifica: risultati (in migliaia di bit)

Risultato della codifica con metodo standard

$$100 * 3 = 300$$

Risultato della codifica con Huffman

$$45 * 1 + 13 * 3 + 12 * 3 + 16 * 3 + 9 * 4 + 5 * 4 = 224$$

Risparmio: 34%

Huffman code

```
HUFFMAN(C)
n = |C|
Q = C
for i = 1 to n − 1
    allocate a new node z
    z.left = x = EXTRACT-MIN(Q)
    z.right = y = EXTRACT-MIN(Q)
    z.freq = x.freq + y.freq
    INSERT(Q, z)
return EXTRACT-MIN(Q)   // root of the tree
```

LZ77 e LZ78

LZ77 e LZ78 sono algoritmi di compressione lossless (senza perdita di informazioni) pubblicati da Abraham Lempel e Jacob Ziv rispettivamente nel 1977 e nel 1978. Questi algoritmi sono alla base di molte varianti come LZW o LZSS. Il metodo trova impiego della compressione di dati eterogenei, testi o immagini, e non necessita di informazioni a priori sui dati da comprimere.

LZ77 e LZ78

L'idea alla base dell'LZ77 è realizzare un dizionario delle parti (token) di dati già incontrati. L'algoritmo di codifica sostituisce i dati già presenti nel dizionario con un riferimento ad essi. Per i primi decenni dalla sua introduzione è stato coperto da brevetti negli Stati Uniti che ne hanno pregiudicato il largo utilizzo, benché fosse popolare sin dalla sua apparizione. La forma più popolare di compressione LZ78 rimane LZW, una variante realizzata da Terry Welch nel 1984 ed utilizzata nei file grafici GIF

LZ78

$$a\ b\ a\ b\ a\ a\ a\ a\ a\ b\ b\ a\ a\ a\ a\ a\ a\ b\ b\ b\ a\ldots$$

$$a|b|a\ b|a\ a|a\ a\ a|b\ b|a\ a\ a\ a|a\ a\ b|b\ b\ a\ldots$$

1. a (-,a)
2. b (-,b)
3. ab (1,b)
4. aa (1,a)
5. aaa (4,a)
6. bb (2,b)
7. $aaaa$ (5,a)
8. aab (4,b)
9. bba (6,a)

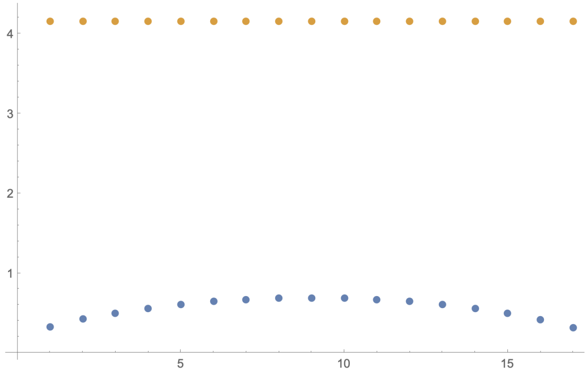
Entropia dei file compressi

Dopo aver codificato la sorgente di informazione per comprimere i dati, otteniamo un file statisticamente simile a un file random (abbiamo eliminato la ridondanza di rappresentazione dell'informazione)

Entropia dei file compressi

```
r1 = {}; r2 = {};
For[i = 0.1, i ≤ 0.9, i = i + .05,
  stringarandom = GeneraStringaBinaria[100 000, i];
  stringacompressa = Compress[stringarandom];
  r1 = Append[r1, Entropy[stringarandom] // N];
  r2 = Append[r2, Entropy[stringacompressa] // N];
 ];
```

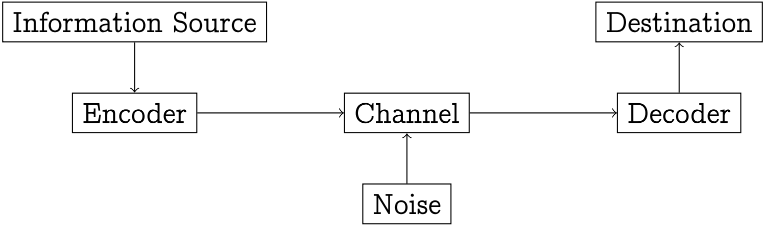
Entropia dei file compressi



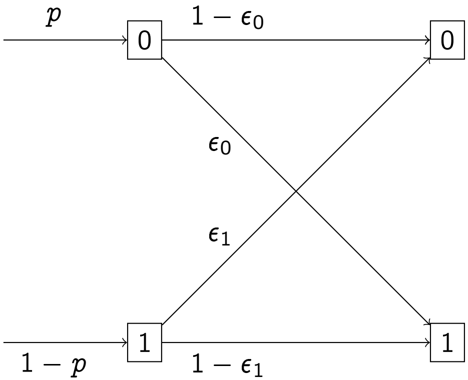
Codifica del canale di comunicazione

A questo punto si procede con la codifica del canale per proteggere l'informazione (molto fragile non avendo più nessuna ridondanza a proteggerla)
Si utilizzano codici a correzione di errore

Teoria dell'Informazione: Scenario



Canale comunicazione



Correzione degli errori

Σ	p_i	w_i
0	p	0
1	$1-p$	1

Probabilità di errore: $p_{err} = p\,\epsilon_0 + (1-p)\epsilon_1$

Se $p = 1/2$ e $\epsilon_0 = \epsilon_1 = \epsilon$: $p_{err} = \epsilon$

Correzione degli errori

Σ	p_i	w_i
0	p	$0 \cdots 0$
1	$1 - p$	$1 \cdots 1$

Probabilità di errore:

quando $|w_i|$ cresce, p_{err} tende a zero

si possono correggere errori in fase di decodifica ma tende a zero anche la velocità di comunicazione !

Correzione degli errori

Obiettivo: correggere errori in decodifica e mantenere alta la velocità di comunicazione

Correzione degli errori

Decodifica con il criterio della minima distanza
decodifica = parola di codice più "vicina" all'output del canale

Distanza 1

$$w_1 = 000$$
$$w_2 = 001$$
$$w_3 = 010$$
$$w_4 = 011$$
$$w_5 = 100$$
$$w_6 = 101$$
$$w_7 = 110$$
$$w_8 = 111$$

Distanza 2 (bit di parità)

$$w_1 = 000$$
$$w_2 = 011$$
$$w_3 = 101$$
$$w_4 = 110$$

Distanza 3

$$
\begin{aligned}
w_1 &= 00000 \\
w_2 &= 00111 \\
w_3 &= 11011 \\
w_4 &= \ldots\ldots
\end{aligned}
$$

Teorema

Un codice binario con distanza $2e+1$ può correggere fino a e errori di trasmissione

Teorema fondamentale della Teoria dell'Informazione

Dato un canale con capacità C, un numero $R < C$ e una probabilità di errore $\epsilon > 0$
esiste $n > 0$ e un codice con 2^{nR} parole di lunghezza n e con probabilità di errore minore di ϵ
Si può trasmettere a qualunque "rate" minore della capacità del canale con errore arbitrariamente basso